Block 3: Learning BN Parameters

1. Introduction.
2. The Noisy OR-Gate Model.
   a. The Basic Noisy OR-Gate Model.
   b. The Leaky Noisy OR-Gate Model.
3. The MLE approach.
   a. Learning a single parameter in a BN with a single node.
   b. Learning multiple parameters in a BN with a single node.
   c. Learning multiple parameters in a BN with many nodes.
   d. Learning with missing data items.
4. Learning parameters with bnlearn.

1. Introduction

So far we have simply shown the conditional probability distributions (**parameters**) in the Bayesian networks we have presented.
But we have not been concerned with how we obtained them!
They can either be obtained:

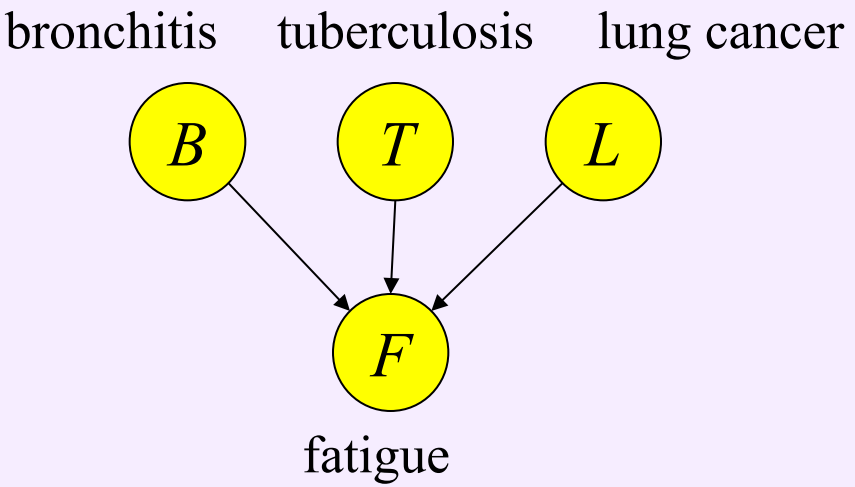
- **From the subjective judgements of an expert in the area (difficult in the case of large networks).**
- **From data.**

Once a model is specified with its parameters, and data have been collected, one is in a position to evaluate its goodness of fit, that is, how well it fits the observed data. Goodness of fit is assessed by finding parameter values of a model that best fits the data: a procedure called **parameter estimation** or **learning parameters**.
We will discuss some techniques for learning parameters from data.

First, we show a technique for simplifying the process of ascertaining the parameters for a node with multiple parents.

2. The Noisy OR-Gate Model

Suppose lung cancer, bronchitis, and tuberculosis all cause **fatigue**, and we need to model this relationship as a part of a system for medical diagnosis.
The portion of the DAG concerning only these four variables is:



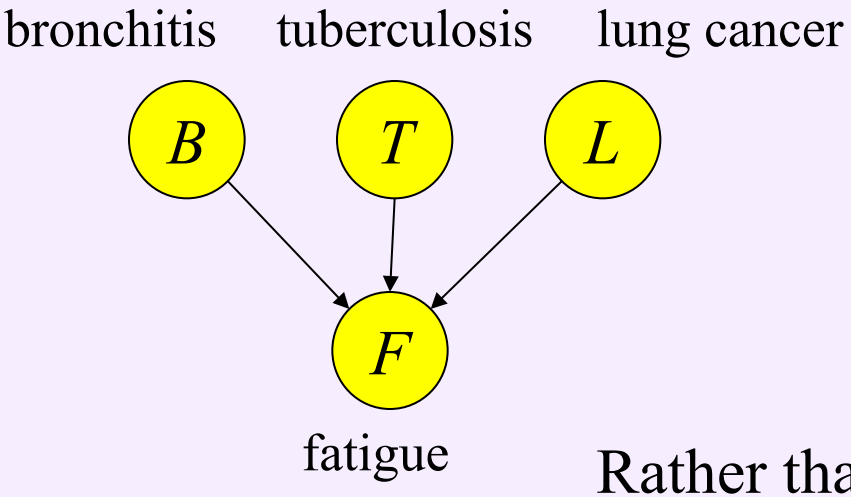
We need to assess eight conditional probabilities for node F:
P(F=yes/B=no, T=no, L=no),..., P(F=yes/B=yes, T=yes, L=yes)

It would be quite difficult to obtain these values from data or from an expert physician. Example: to obtain P(F=yes/B=yes, T=yes, L=no) we need a sufficiently large population of individuals with both bronchitis and tuberculosis, but not lung cancer.

2.a. The Basic Noisy OR-Gate Model

A method for obtaining these conditional probabilities in an indirect way,

1. when the relationships between variables represent causal influences,
2. and each variable has only two values.

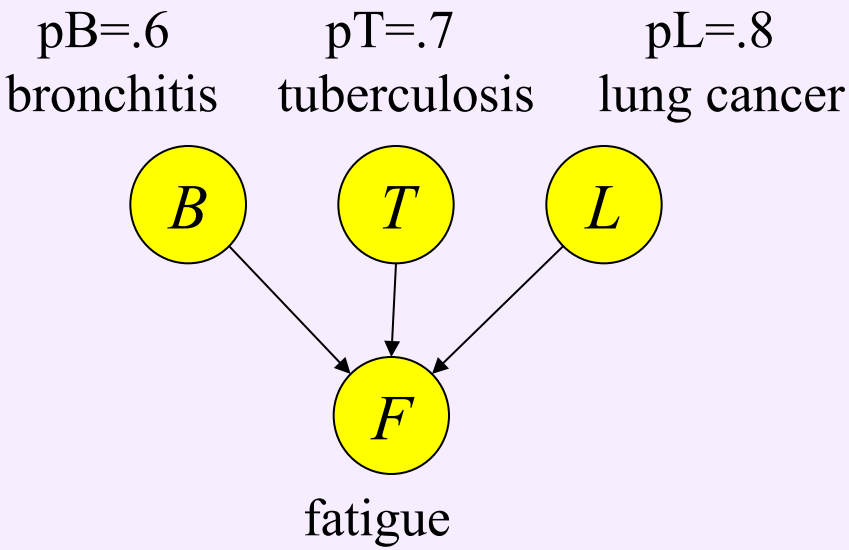


We need to assess eight conditional probabilities for node F:
P(F=yes/B=no, T=no, L=no),...,
P(F=yes/B=yes, T=yes, L=yes)

Rather than assessing all eight probabilities, we assess the **causal strength** of each cause for its effect.

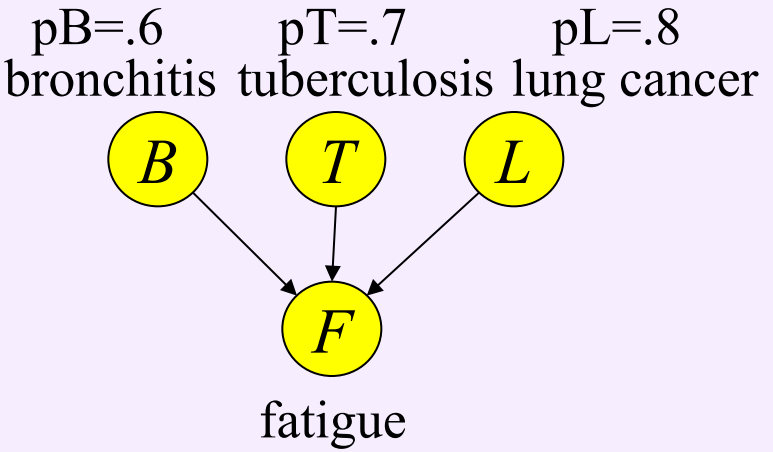
The causal strength is the probability of the cause resulting in the effect whenever the cause is present.

We have shown the causal strength pB of bronchitis for fatigue be 0.6. We assume that bronchitis will always result in fatigue unless some unknown mechanism inhibits this from taking place, and this inhibition takes place 40% of the time. So **60% of the time, bronchitis will result in fatigue**.



Assume that all causes of the effect "fatigue" are articulated in the DAG, and the effect cannot occur unless at least one of its causes is present.

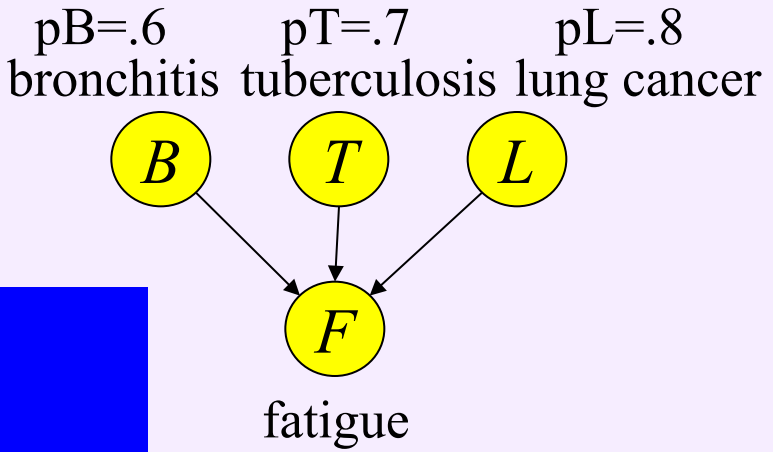
In this case, we have
P(F=yes/B=yes, T=no, L=no) = pB = 0.6
Analogously,
P(F=yes/B=no, T=yes, L=no) = pT = 0.7
P(F=yes/B=no, T=no, L=yes) = pL = 0.8



These 3 causal strengths should not be as difficult to ascertain as all eight conditional probabilities.

For example, to obtain pB from data we only need a population of individuals who have lung bronchitis and do not have the other diseases, and to obtain pB from an expert, the expert need only ascertain the frequency with which bronchitis gives rise to fatigue.

We can obtain the eight conditional probabilities we need from the three causal strengths if we make one additional assumption:

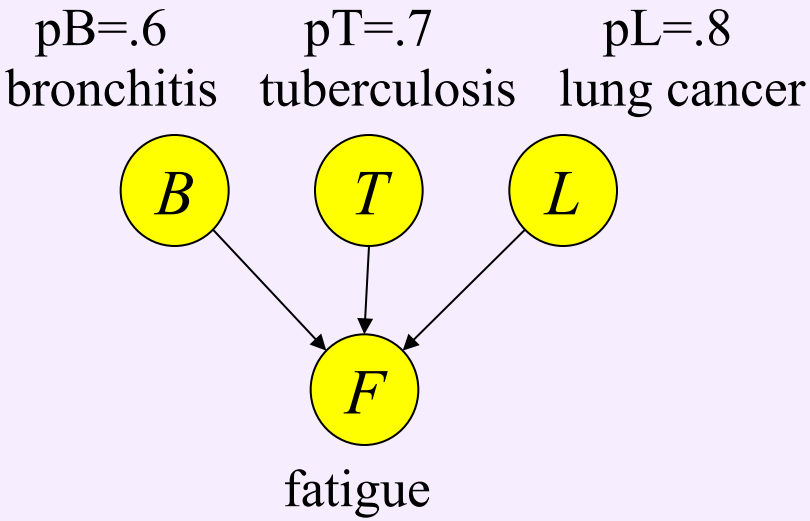


ADDITIONAL ASSUMPTION:

The mechanisms that inhibit the causes act independently from each other.

That is,
$P(F=no/B=yes, T=yes, L=no) = (1-pB)(1-pT) = (1-0.6)(1-0.7)=0.12$

(fatigue should occur unless the causal effects of bronchitis and tuberculosis are both inhibited. Since we assume these inhibitions act independently, the probability that both effects are inhibited is the product of the probabilities that each is inhibited).



Analogously,

$P(F=no/B=yes, T=yes, L=yes) = (1-pB)(1-pT)(1-pL)$

$= (1-0.6)(1-0.7)(1-0.8)=0.024$

Notice that when **more causes** are present, it is **less probable** that fatigue will be absent!

Exercise: obtain the rest of the eight conditional probabilities of this example (since the variables are binary, these are the only values we need to ascertain).

Summarizing, the **noisy OR-gate model** makes the following three assumptions:

① **Causal inhibition**: there is some mechanism which inhibits a cause from bringing about the effect, and the presence of the cause results in the presence of the effect if and only if this mechanism is disabled (turned off).

① **Exception independence**: the mechanism that inhibits one cause is independent of the mechanism that inhibits other causes.

① **Accountability**: an effect can happen only if at least one of its causes is present and is not being inhibited.

The general formula for the **noisy OR-gate model:**

Suppose Y has n causes X1,…, Xn and all variables are binary. Let pi be the causal strength of Xi for Y, that is,

pi=P(Y=yes/X1=no, X2=no,…, Xi=yes,…, Xn=no).

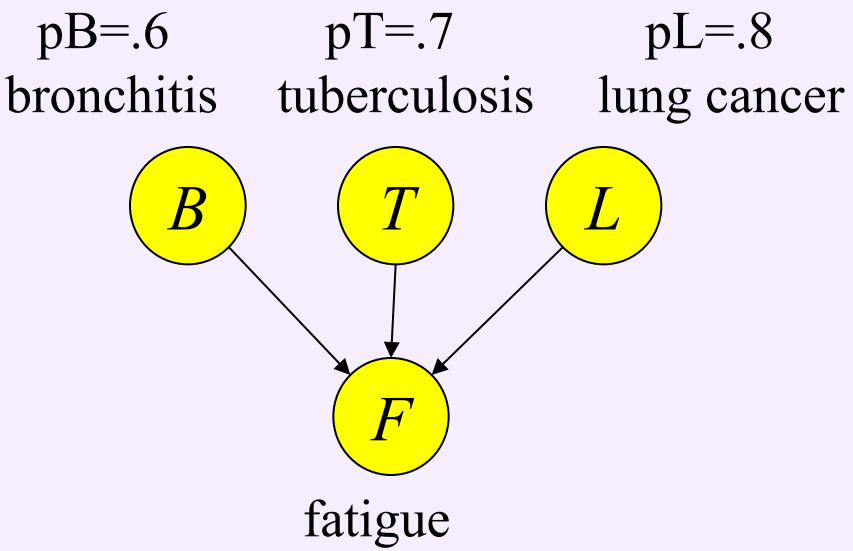
Then, if **A** is a set of nodes that are instantiated to "**yes**",

$$P(Y = no/\mathbf{A}) = \prod_{i \text{ such that } X_i \in \mathbf{A}} (1 - p_i)$$

2.b. The Leaky Noisy OR-Gate Model

Of the three assumptions in the noisy OR-gate model, the **assumption of accountability** seems to be justified less often.

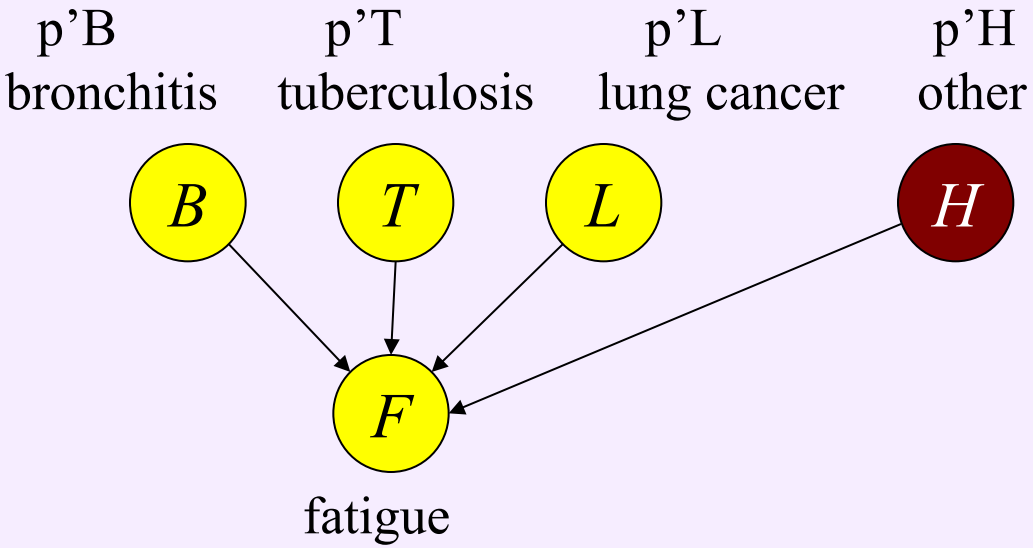
For example, in the case of fatigue, there are certainly other causes of fatigue. So the model of the DAG does not contain all causes of fatigue, and the assumption of accountability is not justified.



In many, if not most, situations, we would not be certain that we have elaborated all known causes of an effect!

Now we show a version of the model that does not assume **accountability: the leaky noisy OR-gate model.**

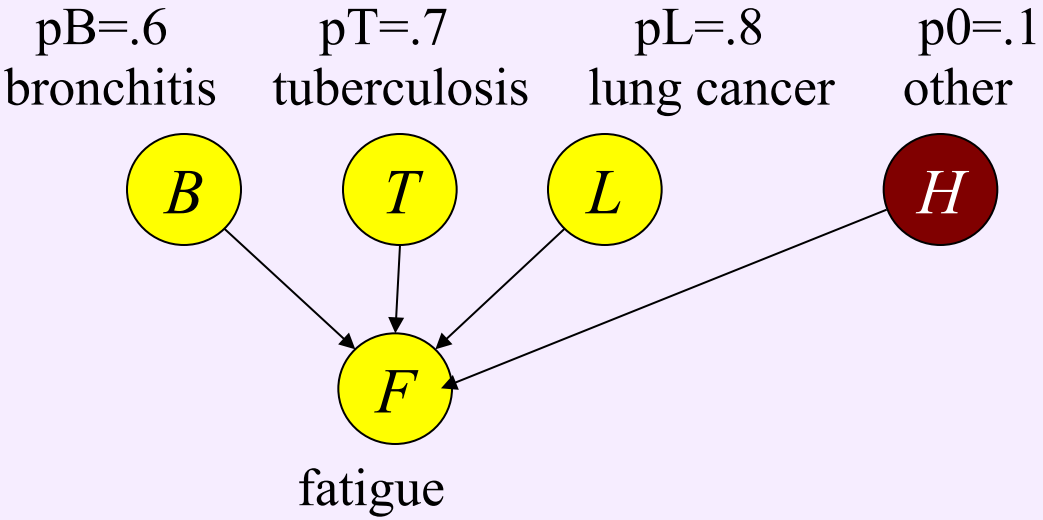
This model assumes that all causes that have not been articulated can be grouped into one other cause *H*, and that the articulated causes, along with *H*, satisfy the three assumptions.



p'B=P(F=yes/B=yes, T=no, L=no, H=no), and p'T, p'L, p'H similar.

We could not ascertain these values because we do not know whether or not H is present!

The only probabilities we could ascertain are:



Recall that pB=P(F=yes/B=yes, T=no, L=no) and that pB does not condition on a value of *H* (while p'B does).

p0 is defined in a different way: p0=P(F=yes/B=no, T=no, L=no) is the probability that the effect will be present given NONE if the articulated causes are present. Note that we are **not** conditioning on *H*.

Goal: develop conditional probability distributions for the BN containing the nodes B, T, L and F from pB, pT, pL and p0.

The general formula for the **leaky noisy OR-gate model:**

Suppose Y has n causes X1,…, Xn and all variables are binary. Let pi be the causal strength of Xi for Y, that is,

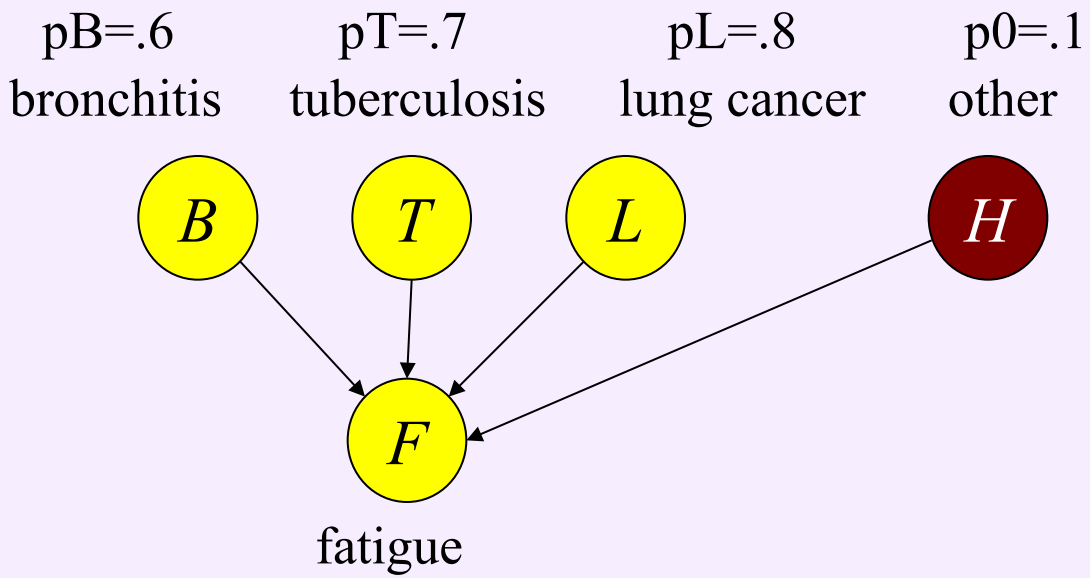
pi=P(Y=yes/X1=no, X2=no,…, Xi=yes,…, Xn=no),

and let p0 the following probability:

p0=P(Y=yes/X1=no,…,Xn=no).

Then, if **A** is a set of nodes that are instantiated to "**yes**",

$$P(Y = no/\mathbf{A}) = (1 - p_0) \prod_{i \text{ such that } X_i \in \mathbf{A}} \frac{1 - p_i}{1 - p_0}$$



P(F=no/B=no, T=no, L=no) = 1−p0 = 1− 0.1 = 0.9
P(F=no/B=no, T=no, L=yes) = (1−p0) (1−pL)/(1−p0)=1−pL=1−0.8=0.2
…

Exercise: Find the rest of the eight conditional probabilities of this example.

3. The MLE approach

Maximum Likelihood Estimation (MLE) is a standard approach to parameter estimation and inference in statistics in general, an in Bayesian Networks in particular.

Let (Γ, P) a BN with $\Gamma=(V, E)$ and $V=\{X1,\ldots,Xn\}$ (we assume the r.v. are discrete)

Let $\mathbf{x}=(x_1,\ldots,x_n)$ be a data vector corresponding to r.v. $X_1,\ldots,X_n$, and let $D=\{\mathbf{x}_1,\ldots,\mathbf{x}_M\}$ be a set of M different data vectors (*"cases"*).

The goal is now to identify the population (i.e. the joint probability distribution) that is most likely to have generated the observed data. Associated with the joint probability distribution is a vector of parameters θ. The MLE of θ is the value that is most likely to have generated the data set D.

3.a. Learning a single parameter in a BN with a single node.

Suppose we picked a coin at random and tossed it. Suppose that we have tossed it 20 times, and it landed heads 18 times. We want to estimate the probability of landing heads from this training data.

Let X_1=*"Side"* be a r.v. whose values are the outcome of the toss (*heads*, 1 or *tails,* 0). Define the parameter

$$\theta = P(Side = heads) = P(X_1 = 1)$$

We can estimate this parameter by the MLE procedure. For that, we consider the associated likelihood function $L(\theta)$ and choose the value of θ that maximizes it.

Likelihood function

Given the observed *training data* $D=\{\mathbf{x}_1,\ldots,\mathbf{x}_M\}$, consisting on M cases, the likelihood function is defined by:

$$L(\theta) = P(D/\theta) = \prod_{i=1}^{M} P(\mathbf{x}_i/\theta)$$

(the likelihood of D being generated from model θ is the product of the probabilities of each case **x**i, given the model)

Maximum Likelihood Estimator (MLE)

Estimates θ by $\boldsymbol{\theta_{MLE}=\arg\max_{\theta} \ln L(\theta)}$, and then approximates the probability of a new case **x** given the training data D as

$$P(\mathbf{x} / D) \approx P(\mathbf{x} / \theta_{MLE})$$

MLE does not assume any **prior** distribution of the parameters θ.

$$L(\theta) = P(X_1^1 = x_1, \ldots, X_1^M = x_M/\theta)$$

Where the observed *training data* D={$\mathbf{x}_1$,…, $\mathbf{x}_M$}, consists on M=20 cases, and each case x_j corresponds to an independent copy of the single r.v. X_1, so x_j is 1 or 0. We know that there is 18 heads "1", and 2 tails "0" in the training data. Let denote by s the number of heads and by t the number of tails. So, s=18, t=2 and M=s+t=20.

$X_1^1, \ldots, X_1^M$ denote the M independent copies of r.v. X1. Then,

$$L(\theta) = P(X_1^1 = x_1, \ldots, X_1^M = x_M/\theta) = \prod_{i=1}^{M} P(X_1^i = x_i/\theta)$$

$$= \prod_{i=1}^{M} \theta^{x_i} (1-\theta)^{1-x_i} = \theta^{\sum_{i=1}^{M} x_i} (1-\theta)^{M-\sum_{i=1}^{M} x_i}$$

$$\ln(L(\theta)) = \ln(\theta) \sum_{i=1}^{M} x_i + \ln(1-\theta)\left(M - \sum_{i=1}^{M} x_i\right)$$

$$\left(\ln(L(\theta))\right)' = \frac{\sum_{i=1}^{M} x_i}{\theta} - \frac{M - \sum_{i=1}^{M} x_i}{1-\theta} = 0$$

$$\hat{\theta} = \frac{\sum_{i=1}^{M} x_i}{M}$$

$$\left(\ln(L(\theta))\right)'' = -\frac{\sum_{i=1}^{M} x_i}{\theta^2} - \frac{M - \sum_{i=1}^{M} x_i}{(1-\theta)^2} < 0$$

$$\text{Then,} \quad \theta_{MLE} = \hat{\theta} = \frac{\sum_{i=1}^{M} x_i}{M}$$

Note that the maximum is **GLOBAL** since the second derivative is always non-positive, not only when substitute θ by the sample mean.

In our case, $\theta_{MLE}=18/20=0.9$ (from data, with no prior distribution for θ)

3.b. Learning multiple parameters in a BN with a single node.

The method just discussed for one single parameter (Bernoulli r.v.) readily extends to multinomial r.v.:

Suppose we are about to repeatedly perform a random process (observe independently a r.v. X_1) with k different outcomes: $(y_1,\ldots, y_k)$.
We assume **exchangeability**, that is, we assign the same probability to all sequences of the same length containing the same number of each outcome.

Let define the parameters

$$\theta_j=P(X_1=y_j),\ j=1,\ldots,k$$

Which are linked through

$$\theta_1+\ldots+\theta_k=1$$

Maximum Likelihood Estimator (MLE)

The observed *training data* $D=\{\mathbf{x}_1,\ldots, \mathbf{x}_M\}$ consists on M cases, and each case x_j corresponds to an independent copy of the single r.v. X_1, so x_j is y_1, y_2,… or y_k. We know that there are n_j values y_j in the training data. So, $n_1+\ldots+n_k=M$.

Then, estimate of $\theta=(\theta_1,\ldots,\theta_k)$ by $\boldsymbol{\theta_{EML}=\arg\max_\theta \ln L(\theta)}$ is obtained similarly to the situation in which k=1:

$$\theta_{j\,MLE} = \frac{n_j}{M}$$

and then we approximate the probability of a new case **x** given the training data D as

$$P(\mathbf{x} / D) \approx P(\mathbf{x} / \theta_{EML})$$

Example:

We have an asymmetrical, six-sided dice, and we have little idea of the probability of each side coming up. We throw the dice. Let X_1 be the outcome of the dice. Let $y_j=j$ for $j=1,\ldots,k=6$, and $\theta_j=P(X_1=j)$ be the parameters.

Suppose next we throw the dice M=100 times, with the following results:

Outcome	Number of occurrences
1	n1=10
2	n2=15
3	n3=5
4	n4=30
5	n5=13
6	n6=27

The **MLE** estimations of θ_j are:

$$\theta_{1\,MLE} = \frac{n_1}{M} = \frac{10}{100} = 0.10$$

$$\theta_{2\,MLE} = \frac{n_2}{M} = \frac{15}{100} = 0.15$$

$$\theta_{3\,MLE} = \frac{n_3}{M} = \frac{5}{100} = 0.05$$

$$\theta_{4\,MLE} = \frac{n_4}{M} = \frac{30}{100} = 0.30$$

$$\theta_{5\,MLE} = \frac{n_5}{M} = \frac{13}{100} = 0.13$$

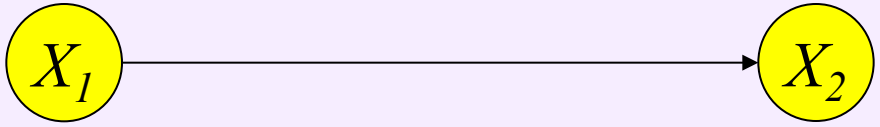
$$\theta_{6\,MLE} = \frac{n_6}{M} = \frac{27}{100} = 0.27$$

3.c. Learning multiple parameters in a BN with many nodes.

Let (Γ, P) a BN with $\Gamma=(V, E)$ and $V=\{X_1,\ldots,X_n\}$ (we assume the r.v. are discrete). For the sake of simplicity, take n=2.

The MLE method of learning parameters in a BN with more than one node follows readily from the method for learning parameters with a single node. We illustrate this method with binomial r.v but it extends readily to the multinomial case. So, we assume by the moment that X_1 and X_2 can only take values 0 and 1.

The parameters of this BN are three: θ_{11}, θ_{21} and θ_{22}:



$\theta_{11}=P(X_1=1)$

$\theta_{21}=P(X_2=1/X_1=1)$

$\theta_{22}=P(X_2=1/X_1=0)$

$\theta_{11}=P(X_1=1)$

$\theta_{21}=P(X_2=1/X_1=1)$
$\theta_{22}=P(X_2=1/X_1=0)$

Let $\mathbf{x}=(x_1,x_2)$ be a data vector corresponding to r.v. X_1, X_2, and let $D=\{\mathbf{x}_1,\ldots,\mathbf{x}_M\}$ be a set of M different data vectors (*"cases"*).

The goal again is to identify the population (i.e. the joint probability distribution) that is most likely to have generated the observed data. We do not assume any a priori distribution of the parameter θ.

The MLE of θ is the value that is most likely to have generated the data set D.

Given the observed *training data* $D=\{\mathbf{x}_1,\ldots, \mathbf{x}_M\}$, consisting on M cases, the likelihood function is:

$$
\begin{aligned}
L(\theta) &= P\big((X_1^1, X_2^1) = (x_1^1, x_2^1), \ldots, (X_1^M, X_2^M) = (x_1^M, x_2^M)/\theta\big) \\
&= \prod_{i=1}^{M} P((X_1^i, X_2^i) = (x_1^i, x_2^i)/\theta) \\
&= \prod_{i=1}^{M} P(X_1^i = x_1^i/\theta)\, P(X_2^i = x_2^i/X_1^i = x_1^i, \theta) \\
&= \prod_{i=1}^{M} \theta_{11}^{x_1^i} (1-\theta_{11})^{1-x_1^i}\, \theta_{21}^{x_2^i\, x_1^i} (1-\theta_{21})^{(1-x_2^i)\, x_1^i} \\
&\qquad \theta_{22}^{x_2^i\, (1-x_1^i)} (1-\theta_{22})^{(1-x_2^i)\, (1-x_1^i)} \\
&= \theta_{11}^{s_{11}} (1-\theta_{11})^{t_{11}}\, \theta_{21}^{s_{21}} (1-\theta_{21})^{t_{21}}\, \theta_{22}^{s_{22}} (1-\theta_{22})^{t_{22}}
\end{aligned}
$$

That is, $$L(\theta) = \theta_{11}^{s_{11}} (1-\theta_{11})^{t_{11}} \theta_{21}^{s_{21}} (1-\theta_{21})^{t_{21}} \theta_{22}^{s_{22}} (1-\theta_{22})^{t_{22}}$$

where:

$$\theta = (\theta_{11}, \theta_{21}, \theta_{22})$$

s_{11} = number of cases for which $X_1 = 1$

t_{11} = number of cases for which $X_1 = 0$

s_{21} = number of cases for which $X_2 = 1$ and $X_1 = 1$

t_{21} = number of cases for which $X_2 = 0$ and $X_1 = 1$

s_{22} = number of cases for which $X_2 = 1$ and $X_1 = 0$

t_{22} = number of cases for which $X_2 = 0$ and $X_1 = 0$

Note that $s_{11} + t_{11} = M$

$$s_{21} + t_{21} + s_{22} + t_{22} = M$$

$$\ln(L(\theta)) = s_{11}\ln(\theta_{11}) + t_{11}\ln(1-\theta_{11}) + s_{21}\ln(\theta_{21}) + t_{21}\ln(1-\theta_{21}) + s_{22}\ln(\theta_{22}) + t_{22}\ln(1-\theta_{22})$$

$$\frac{\partial L(\theta)}{\partial \theta_{11}} = \frac{s_{11}}{\theta_{11}} - \frac{t_{11}}{1-\theta_{11}} = 0 \Rightarrow \hat{\theta}_{11} = \frac{s_{11}}{s_{11}+t_{11}} = \frac{s_{11}}{M}$$

$$\frac{\partial L(\theta)}{\partial \theta_{21}} = \frac{s_{21}}{\theta_{21}} - \frac{t_{21}}{1-\theta_{21}} = 0 \Rightarrow \hat{\theta}_{21} = \frac{s_{21}}{s_{21}+t_{21}} \quad \text{if } s_{21}+t_{21} > 0$$

$$\frac{\partial L(\theta)}{\partial \theta_{22}} = \frac{s_{22}}{\theta_{22}} - \frac{t_{22}}{1-\theta_{22}} = 0 \Rightarrow \hat{\theta}_{22} = \frac{s_{22}}{s_{22}+t_{22}} \quad \text{if } s_{22}+t_{22} > 0$$

In order to check that $\hat{\theta} = (\hat{\theta}_{11}, \hat{\theta}_{21}, \hat{\theta}_{22})$ is the MLE, we must find the Hessian matrix:

$$H(L(\theta)) = \begin{pmatrix} \frac{\partial^2 L(\theta)}{\partial \theta_{11}^2} & \frac{\partial^2 L(\theta)}{\partial \theta_{11}\,\partial \theta_{21}} & \frac{\partial^2 L(\theta)}{\partial \theta_{11}\,\partial \theta_{22}} \\ \frac{\partial^2 L(\theta)}{\partial \theta_{21}\,\partial \theta_{11}} & \frac{\partial^2 L(\theta)}{\partial \theta_{21}^2} & \frac{\partial^2 L(\theta)}{\partial \theta_{21}\,\partial \theta_{22}} \\ \frac{\partial^2 L(\theta)}{\partial \theta_{22}\,\partial \theta_{11}} & \frac{\partial^2 L(\theta)}{\partial \theta_{22}\,\partial \theta_{21}} & \frac{\partial^2 L(\theta)}{\partial \theta_{22}^2} \end{pmatrix}$$

and compute the second partial derivatives:

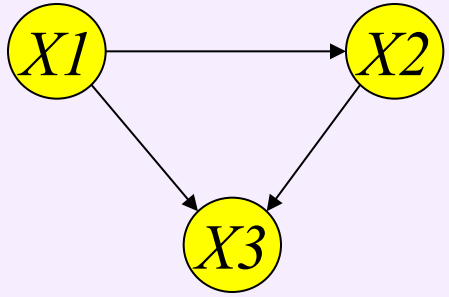
$$H(L(\theta)) = \begin{pmatrix} -\frac{s_{11}}{\theta_{11}^2} - \frac{t_{11}}{(1-\theta_{11})^2} & 0 & 0 \\ 0 & -\frac{s_{21}}{\theta_{21}^2} - \frac{t_{21}}{(1-\theta_{21})^2} & 0 \\ 0 & 0 & -\frac{s_{22}}{\theta_{22}^2} - \frac{t_{22}}{(1-\theta_{22})^2} \end{pmatrix}$$

which is a negative definite matrix for all the values of θ_{11}, θ_{21} and θ_{22}. Then, $L(\theta)$ attains a local maximum (which is indeed a **GLOBAL** maximum) at $\hat{\theta} = (\hat{\theta}_{11}, \hat{\theta}_{21}, \hat{\theta}_{22})$.

Then,

$$\theta_{MLE} = \hat{\theta} = \left(\frac{s_{11}}{M}, \frac{s_{21}}{s_{21}+t_{21}}, \frac{s_{22}}{s_{22}+t_{22}}\right)$$

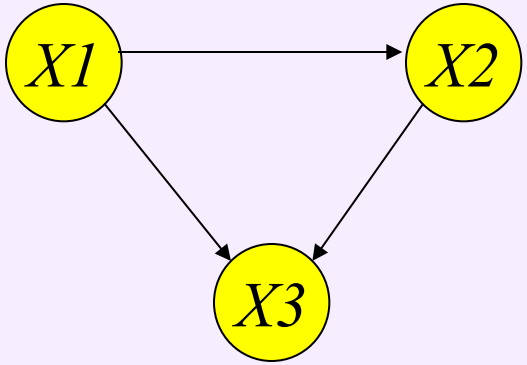
Example:



Suppose we start with the DAG:

and variables X_1, X_2 and X_3 are binary r.v. The parameters are:

$\theta_{11}=P(X1=1)$



$\theta_{21}=P(X2=1/X1=1)$

$\theta_{22}=P(X2=1/X1=0)$

$\theta_{31}=P(X3=1/X1=1, X2=1)$

$\theta_{32}=P(X3=1/X1=0, X2=1)$

$\theta_{33}=P(X3=1/X1=1, X2=0)$

$\theta_{34}=P(X3=1/X1=0, X2=0)$

If we have a set of data D with M cases, denote by:
s11 = the count for variable X1=1.
t11 = the count for variable X1=0.

s21 = the count for X2=1, X1=1.
t21 = the count for X2=0, X1=1.
s22 = the count for X2=1, X1=0.
t22 = the count for X2=0, X1=0.

s31 = the count for X3=1, X1=1, X2=1.
t31 = the count for X3=0, X1=1, X2=1.
s32 = the count for X3=1, X1=0, X2=1.
t32 = the count for X3=0, X1=0, X2=1.
s33 = the count for X3=1, X1=1, X2=0.
t33 = the count for X3=0, X1=1, X2=0.
s34 = the count for X3=1, X1=0, X2=0.
t34 = the count for X3=0, X1=0, X2=0.

Note that
s11 + t11 = M,

s21+ t21 = s11,
s22 + t22 = t11,

s31 + t31 = s21,
s32 + t32 = t21,
s33 + t33 = s22,
s34 + t34 = t22.

Then, the MLE estimation of θ=(θ11, θ21, θ22, θ31, θ32, θ33, θ34) is:

$$\theta_{11,MLE} = \frac{s_{11}}{s_{11}+t_{11}} = \frac{s_{11}}{M}$$
$$\theta_{21,MLE} = \frac{s_{21}}{s_{21}+t_{21}} = \frac{s_{21}}{s_{11}}$$
$$\theta_{22,MLE} = \frac{s_{22}}{s_{22}+t_{22}} = \frac{s_{22}}{t_{11}}$$
$$\theta_{31,MLE} = \frac{s_{31}}{s_{31}+t_{31}} = \frac{s_{31}}{s_{21}}$$
$$\theta_{32,MLE} = \frac{s_{32}}{s_{32}+t_{32}} = \frac{s_{32}}{t_{21}}$$
$$\theta_{33,MLE} = \frac{s_{33}}{s_{33}+t_{33}} = \frac{s_{33}}{s_{22}}$$
$$\theta_{34,MLE} = \frac{s_{34}}{s_{34}+t_{34}} = \frac{s_{34}}{t_{22}}$$

In general:

The parmeters of the Bayesian network are of the form:

$$\Theta_{x/u} = P(X=x \,/\, PA(X)=u)$$

where X represents any node in V and PA(X) is the set of nodes in V which are parents of X. Assume that all the parameters are well defined.

Let Θ denote the vector whose components are the parameters of the model, and let $L(\Theta|D)$ denote the likelihood function associated to the parameter's vector and to the training data set.

The, the MLE (Maximum Likelihood Estimator) is the value of Θ that maximizes the likelihood function (equivalently, the logarithm of the likelihood function), that is,

$$\Theta^{\wedge} = \text{argmax}_{\Theta}\ L(\Theta|D) = \text{argmax}_{\Theta}\ \log(L(\Theta|D)) \ .$$

THEOREM 3 (Th. 17. 1 Darwiche, 2009)

If D, the set of training data, is composed of M **complete** cases (without missing observations), then the MLE $\Theta^{\wedge}$ verifies that its components are obtained from the empirical distribution function, that is,

$$\Theta^{\wedge}_{x/u} = \# D(x,u) / \# D(u)$$

where # D(x,u) is the cardinal of the set of cases in D for which X = x and PA(X) = u, and # D(u) is the cardinal of the set of cases in D for which just PA(X) = u.

By Theorem 3, if the training data set is composed of **complete** cases, the MLE of Θ, $\Theta^{\wedge}$, **EXISTS AND IS UNIC**.

TEOREMA 4 (Th. 17.2 Darwiche, 2009)

If D, the set of training data, is composed of M **complete** cases (without missing observations), then the MLE $\hat{\Theta}$ minimizes the Kullback-Leibler divergence between the empirical distribution of probability P* that assigns to an even its relative frequency in D, and the probability distribution P_Θ associated to the Bayesian network (the joint probability distribution of random variables in V), with vector of parameters Θ, that is,

$$\hat{\Theta} \; (= \text{argmax}_\Theta \; L(\Theta|D) \;)$$
$$= \text{argmin}_\Theta \, KL(P^*, P_\Theta \,)$$

3.d. Learning with missing data items.

So far we have considered data sets in which every value of every variable is recorded in every case. Next we consider the case in which some data items might be omitted by simple **random omissions** due to recording problems or some similar error.

The parameter ML estimation considered previously has a number of interesting properties:

- it is unique,
- it is asymptotically Normal, and
- it maximizes the probability of data.

Most importantly, this estimation is easily computable from a data set. It is therefore common to seek **maximum likelihood estimates** for incomplete data sets.

We present one of the methods that search for ML estimates under incomplete data, which is based on local search, which start with some initial estimates and then iteratively improve on them until some stopping condition is met.

This method is generally more expensive than the method for complete data yet it is not generally guaranteed to find ML estimates! The method is called:

Expectation Maximization (EM).

This method first completes the incomplete data set, and then uses it to estimate parameters as with the MLE approach when we had complete data.
The new set of parameters are guaranteed to have no less likelihood than the initial parameters, so this process can be repeated until some convergence condition is met.

Consider the following Bayesian network:

$\theta_{11}=P(X1=1)$

$\theta_{21}=P(X2=1/X1=1)$

$\theta_{22}=P(X2=1/X1=0)$

Our goal is to find ML estimates for the parameters θ_{11}, θ_{21} and θ_{22}, from the following (incomplete) data set $D=\{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4,\mathbf{x}_5\}$ (M=5)

Case	*X1*	*X2*
1	1	1
2	1	?
3	1	1
4	1	0
5	0	?

That is, $\mathbf{x}_1=(1,1)$, $\mathbf{x}_2=(1,?)$,
$\mathbf{x}_3=(1,1)$, $\mathbf{x}_4=(1,0)$, $\mathbf{x}_5=(0,?)$

Suppose further that we are starting with the initial estimates $\theta_{11}^0 = 0.25\,,\quad \theta_{21}^0 = 0.80\,,\quad \theta_{22}^0 = 0.40$

which have the following likelihood function from data set D:

$$L(\theta^0/D) = \prod_{i=1}^{M=5} P(\mathbf{x}_i/\theta^0)$$

$$= P_{\theta^0}(X_1 = 1,\, X_2 = 1)\, P_{\theta^0}(X_1 = 1)\, P_{\theta^0}(X_1 = 1,\, X_2 = 1)\, P_{\theta^0}(X_1 = 1,\, X_2 = 0)\, P_{\theta^0}(X_1 = 0)$$

$$= (\theta^0_{21}\,\theta^0_{11}) \times \theta^0_{11} \times (\theta^0_{21}\,\theta^0_{11}) \times \left((1-\theta^0_{21})\,\theta^0_{11}\right) \times (1-\theta^0_{11})$$

$$= 0.80 \times 0.25 \times 0.25 \times 0.80 \times 0.25 \times 0.20 \times 0.25 \times 0.75$$

$$= 0.000375$$

Note that when the data is **complete**, each term in this product can be evaluated using the chain rule for Bayesian networks. Contrary to this case, when data is **incomplete** evaluating the terms in this product generally requires inference on the Bayesian network.
For example, imagine that one case were **x**=(?,0). Then, the corresponding term in the product of the likelihood function will be:

$$P_{\theta^0}(X_2 = 0) = P_{\theta^0}(X_2 = 0/X_1 = 1)\, P_{\theta^0}(X_1 = 1) + P_{\theta^0}(X_2 = 0/X_1 = 0)\, P_{\theta^0}(X_1 = 0)$$

$$= (1-\theta^0_{21}) \times \theta^0_{11} + (1-\theta^0_{22}) \times (1-\theta^0_{11})$$

$$= 0.20 \times 0.25 + 0.60 \times 0.75 = 0.5$$

Case	X1	X2
1	1	1
2	1	?
3	1	1
4	1	0
5	0	?

First of all we need to complete the incomplete data set D, and then use it to estimate parameters as with the MLE approach with complete data.

In this case, there are four possible completations:

Case	X1	X2
1	1	1
2	1	1
3	1	1
4	1	0
5	0	1

Case	X1	X2
1	1	1
2	1	1
3	1	1
4	1	0
5	0	0

Case	X1	X2
1	1	1
2	1	0
3	1	1
4	1	0
5	0	1

Case	X1	X2
1	1	1
2	1	0
3	1	1
4	1	0
5	0	0

Although we do not know which one of these completations is the correct one, we can compute the probability of each completation based on the initial set of parameters we have:

Completations			
Case	*X1*	*X2*	**P0(/ xi)**
1	1	1	**1**
2	1	**1**	**$P_0(X2=1/X1=1)=\theta 21=0.80$**
2	1	**0**	**$P_0(X2=0/X1=1)=1-\theta 21=0.20$**
3	1	1	**1**
4	1	0	**1**
5	0	**1**	**$P_0(X2=1/X1=0)=\theta 22=0.40$**
5	0	**0**	**$P_0(X2=0/X1=0)=1-\theta 22=0.60$**

Probability of the variables with missing values in case xi, under $\theta 0$.

From this table we can obtain an expected empirical distribution for the variables X_1 and X_2:

Completations			
Case	*X1*	*X2*	P0(/ **x**i)
1	1	1	**1**
2	1	**1**	**0.80**
2	1	**0**	**0.20**
3	1	1	**1**
4	1	0	**1**
5	0	**1**	**0.40**
5	0	**0**	**0.60**

Expected empirical distribution:

X1	*X2*	
1	1	0.56
1	0	0.24
0	1	0.08
0	0	0.12
		+1

(1+0.80+1)/5=0.56

(0.20+1)/5=0.24

0.40/5=0.08

0.60/5=0.12

And now we can use this expected empirical distribution to estimate the parameters in the following way:

$$\hat{\theta}^1_{11} = 0.56 + 0.24 = 0.80$$

$$\hat{\theta}^1_{21} = \frac{0.56}{0.56 + 0.24} = \frac{0.56}{0.80} = 0.7$$

$$\hat{\theta}^1_{22} = \frac{0.08}{0.08 + 0.12} = \frac{0.08}{0.20} = 0.4$$

$$L(\theta^1/D) = \prod_{i=1}^{M=5} P(\mathbf{x}_i/\theta^1)$$

$$= P_{\theta^1}(X_1 = 1,\, X_2 = 1)\, P_{\theta^1}(X_1 = 1)\, P_{\theta^1}(X_1 = 1,\, X_2 = 1)\, P_{\theta^1}(X_1 = 1,\, X_2 = 0)\, P_{\theta^1}(X_1 = 0)$$

$$= (\theta^1_{21}\,\theta^1_{11}) \times \theta^1_{11} \times (\theta^1_{21}\,\theta^1_{11}) \times \left((1-\theta^1_{21})\,\theta^1_{11}\right) \times (1-\theta^1_{11})$$

$$= 0.70 \times 0.80 \times 0.80 \times 0.70 \times 0.80 \times 0.30 \times 0.80 \times 0.20$$

$$= 0.01204224\ (> L(\theta^0/D) = 0.000375\,)$$

Hence, the new estimates have a higher likelihood than the initial ones we started with:

Iteration	$i = 0$	$i = 1$
θ^i_{11}	0.25	0.80
θ^i_{21}	0.80	0.70
θ^i_{22}	0.40	0.40
$L(\theta^i/D)$	0.000375	0.01204224

An then we can now compute again the probability of each completation based on the new estimation of the parameters we have:

Completations			
Case	*X1*	*X2*	**P1(/ xi)**
1	1	1	**1**
2	1	**1**	**$P_1(X2=1/X1=1)=\theta 21=0.70$**
2	1	**0**	**$P_1(X2=0/X1=1)=1-\theta 21=0.30$**
3	1	1	**1**
4	1	0	**1**
5	0	**1**	**$P_1(X2=1/X1=0)=\theta 22=0.40$**
5	0	**0**	**$P_1(X2=0/X1=0)=1-\theta 22=0.60$**

Probability of the variables with missing values in case xi, under $\theta 1$.

From this table we can obtain a new expected empirical distribution for the variables X_1 and X_2:

Completations			
Case	*X1*	*X2*	P1(/ **x**i)
1	1	1	**1**
2	1	**1**	**0.70**
2	1	**0**	**0.30**
3	1	1	**1**
4	1	0	**1**
5	0	**1**	**0.40**
5	0	**0**	**0.60**

Expected empirical distribution:

X1	*X2*	
1	1	0.54
1	0	0.24
0	1	0.08
0	0	0.12
		+1

(1+0.70+1)/5=0.54

(0.30+1)/5=0.26

0.40/5=0.08

0.60/5=0.12

And now we can use this expected empirical distribution to find the new estimate of the parameters:

$$\hat{\theta}^2_{11} = 0.54 + 0.26 = 0.80$$

$$\hat{\theta}^2_{21} = \frac{0.54}{0.54 + 0.26} = \frac{0.54}{0.80} = 0.675$$

$$\hat{\theta}^2_{22} = \frac{0.08}{0.08 + 0.12} = \frac{0.08}{0.20} = 0.40$$

$$L(\theta^2/D) = \prod_{i=1}^{M=5} P(\mathbf{x}_i/\theta^2)$$

$$= P_{\theta^2}(X_1 = 1,\, X_2 = 1)\, P_{\theta^2}(X_1 = 1)\, P_{\theta^2}(X_1 = 1,\, X_2 = 1)\, P_{\theta^2}(X_1 = 1,\, X_2 = 0)\, P_{\theta^2}(X_1 = 0)$$

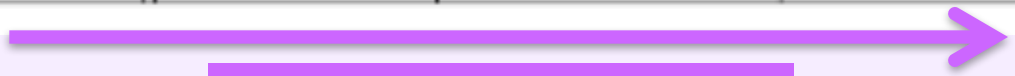
$$= (\theta^2_{21}\,\theta^2_{11}) \times \theta^2_{11} \times (\theta^2_{21}\,\theta^2_{11}) \times \left((1-\theta^2_{21})\,\theta^2_{11}\right) \times (1-\theta^2_{11})$$

$$= 0.675 \times 0.80 \times 0.80 \times 0.675 \times 0.80 \times 0.325 \times 0.80 \times 0.20$$

$$= 0.01213056\ (> L(\theta^1/D) = 0.01204224\,)$$

Hence, the new estimates have a higher likelihood than the previous ones:

Iteration	$i = 0$	$i = 1$	$i = 2$
θ^i_{11}	0.25	0.80	0.80
θ^i_{21}	0.80	0.70	0.675
θ^i_{22}	0.40	0.40	0.40
$L(\theta^i/D)$	0.000375	0.01204224	0.01213056



Increasing likelihood

Iteration	$i = 0$	$i = 1$	$i = 2$
$\theta_{11}^i = P(X_1 = 1)$	0.25	0.80	0.80
θ_{21}^i	0.80	0.70	0.675
θ_{22}^i	0.40	0.40	0.40
$L(\theta^i/D)$	0.000375	0.01204224	0.01213056
$P(X_2 = 1)$	0.5	0.64	0.62

$$P(X_2 = 1) = P(X_2 = 1/X_1 = 1)\, P(X_1 = 1) + P(X_2 = 1/X_1 = 0)\, P(X_1 = 0)$$
$$= \theta_{21}\, \theta_{11} + \theta_{22}\, (1 - \theta_{11})$$

Note: the algorithm finishes after some prefixed number of iterations or when the differences between successive evaluations of the likelihood function (or of the logarithm of the likelihood function) are very small.

This algorithm is known as the "ML_EM (Maximum Likelihood Expectation Maximization) algorithm".

- This algorithm converges to a local maximum of the likelihood function, but the algorithm may converge to different estimations of the parameters with different likelihoods depending on the initial estimates θ_0.

- It is therefore not uncommon to run the algorithm multiple times, starting with different estimates in each iteration (perhaps chosen randomly) and then returning the best estimates found across all iterations.

- The EM algorithm is known to converge very slowly if the fraction of missing data is quite large. It is sometimes sped up using the gradient ascent approach, which we do not introduce in this course.

4. Learning parameters with bnlearn

Recall that we saw there exist different ways of creating a fitted BN (a bn object with parameters: bn.fit object):

1. Expert-driven approach: in which the parameters are specified by the user using custom.fit(), which takes a bn object encoding the network structure and a list with the parameters of the local distributions of the nodes. **For this approach we prefer gRain!**

Now we will consider the following two approaches:

2. Data-driven approach: learning it from a data set using bn.fit() and a network structure (bn object).
3. Hybrid approach: combining the above (using the assignment operator for bn.fit objects, which replaces the parameters of a single local distribution).

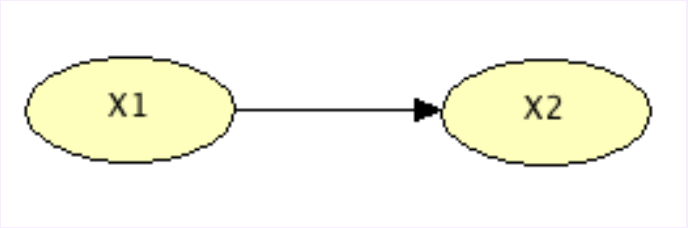
And introduce them through an example:

Consider the Bayesian Network with this DAG. Variables X_1 and X_2 take values true/false. The parameters are:

$\theta_{11}=P(X_1=true)=0.25$

$\theta_{21}=P(X_2=true/X_1=true)=0.80$

$\theta_{22}=P(X_2=true/X_1=false)=0.40$

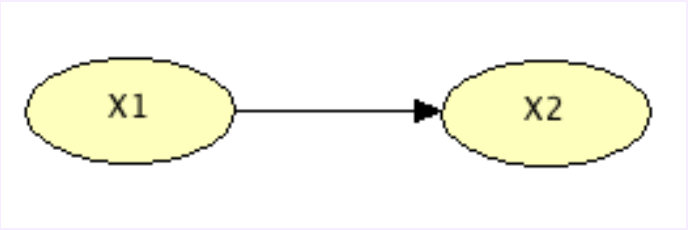


He have generated a random set of 500 cases for this Bayesian Network (following the joint probability distribution P given by the CPT of X_2 conditioned to X_1, and the probability distribution of X_1, by the Chain rule.

This data set is named "Ejemplo_seccion4.rdata" and when you load it with R you will have a dataframe with name "**ejemplo_1**".

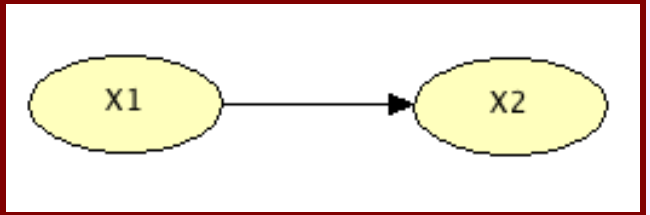
If we had forgotten the values of the parameters, we could apply learning parameters with bnlearn to learn the parameters from the data (and then we can compare with the actual values!).

With bnlearn, first, we have to create a bn object with the graph structure of the Bayesian network. In this case, we will create it "by hand". In Block 4 we will see how to create by learning from data: structure learning.

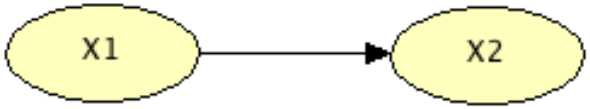



```
library(bnlearn)
library(Rgraphviz)
# Now we create a empty network:
names<-c("X1","X2")
net=empty.graph(names)
net
class(net)
# we see that "net" is a bn object.
```

```
# Now introduce the arrows:
arcs(net,ignore.cycles=TRUE)=
matrix(c("X1","X2"), ncol=2, byrow=TRUE,
dimnames=list(c(),c("from","to")))
#
net
# The graph:
plot1<-graphviz.plot(net)
plot1
#############################################################
# Alternatively, we could introduce arrows in this way:
mat=matrix(c(0,1,0,0),nrow=2,byrow=TRUE,
dimnames=list(nodes(net),nodes(net)))
mat
net2=empty.graph(names)
amat(net2)=mat
all.equal(net,net2)
```



```
# Or even…
net3=empty.graph(names)
# or
net3=empty.graph(nodes(net))
#
net3=set.arc(net3, "X1","X2")
all.equal(net,net3)
##########################################################################
# Now we load the data randomly generated from the Bayesian
network: data frame “ejemplo_1”.
# We estimate the probabilities constructing an object bn.fit:
net.estimated=bn.fit(net, ejemplo_1, method="mle")
net.estimated
class(net.estimated)
# We can save coefficients in a list named “coef”:
coef<-coefficients(net.estimated)
str(coef)
```



```
> coef[1]
$X1
      false  true
0.042 0.740 0.218
```

Normalizing

X_1	False (0)	True (1)
	0.7724425887	$\theta_{11} = 0.2275574113$

```
> #
> coef[2]
$X2
       X1
X2                      false       true
       0.00000000 0.04324324 0.05504587
  false 0.28571429 0.54594595 0.16513761
  true  0.71428571 0.41081081 0.77981651
```

Normalizing

$X_1 \rightarrow$ $X_2 \downarrow$	False (0)	True (1)
False (0)	0.5706214712	0.1747572782
True (1)	$\theta_{22} = 0.4293785288$	$\theta_{21} = 0.8252427218$

```
# Estimations of the parameters are, then, by normalizing:
# theta_11=P(X1=true)=
theta_11=coef$X1[3]/(1-coef$X1[1])
# theta_21=P(X2=true/X1=true)
theta_21=coef$X2[3,3]/(1-coef$X2[1,3])
# theta_22=P(X2=true/X1=false)
theta_22=coef$X2[3,2]/(1-coef$X2[1,2])
#
theta<-cbind(theta_11,theta_21,theta_22)
theta
```

```
> theta
      theta_11  theta_21  theta_22
true 0.2275574 0.8252427 0.4293785
```

```
#
#
P_X2_true<-theta_21*theta_11+theta_22*(1-theta_11)
P_X2_true
```

```
> P_X2_true
     true
0.5194604
```

```
#
#
```

Summary: Evaluating ML_EM algorithm.

Parameters	Actual values	bnlearn
$\theta_{11} = P(X_1 = true)$	0.25	0.2275574113 (−0.0224425887)
$\theta_{21} = P(X_2 = true/X_1 = true)$	0.80	0.8252427218 (+0.0252427218)
$\theta_{22} = P(X_2 = true/X_1 = false)$	0.40	0.4293785288 (+0.0293785288)
$P(X_2 = 1)$	0.5	0.5194603598 (+0.0194603598)